



Short Communication

A note on a decision support system for multiobjective integer and mixed-integer programming problems

Maria João Alves^{*}, João Clímaco

Faculdade de Economia, Universidade de Coimbra/INESC COIMBRA, Av. Dias da Silva, 165, 3004-512 Coimbra, Portugal

Received 11 October 2002; accepted 11 October 2002

Abstract

This note presents the main characteristics of a decision support system (DSS) dealing with multiobjective integer and mixed-integer programming problems. The DSS is based on interactive reference point approaches developed by the authors for this kind of problems. It is implemented for Windows platforms and aims at providing an open communication protocol for interaction with the decision maker(s).

© 2003 Elsevier B.V. All rights reserved.

Keywords: Multiple objective programming; Decision support systems; Integer linear programming

1. Introduction

Some interactive decision support systems (DSSs) have been developed for dealing with multiple objective integer (or mixed-integer) linear programming problems (MOILP/MOMILP). They implement interactive procedures, which consider distinct protocols to interact with the decision maker (DM) and to guide the decision process, and use different techniques to compute nondominated solutions. Examples of different interactive approaches are described in [5–7].

In our opinion, open communication approaches, which enable a free exploration of the

problem and a progressive learning of the non-dominated solution set, are more suitable for dealing with MOILP/MOMILP problems than those approaches that aim at converging to the optimum of an implicit utility function. In open communication approaches there are no irrevocable decisions during the whole process and the DM is always allowed to go *backwards* at a later interaction. For instance, finding twice the same solution does not necessarily mean entering in cycle. The state of knowledge of the DM can vary during this learning process, and so different options originating from the same starting solution may occur. The process only finishes when the DM considers to have gained sufficient insight into the nondominated solution set to base a final decision.

We have developed an open communication DSS whose principal module includes the interactive reference point procedures [1,2], previously

^{*} Corresponding author. Tel.: +351-39-790-500; fax: +351-39-403-511.

E-mail address: mjoao@inescc.pt (M.J. Alves).

developed for MOILP/MOMILP problems. Since we believe that joining procedures, as pieces of a DSS, may not be enough to build an effective DSS, we have developed the methodology underlying the DSS and simultaneously we have investigated the way it should be integrated into a computational system. The interactive procedures underlying the system provide a simple protocol to interact with the DM (not demanding too much information about his/her preferences at each interaction), allowing the DM to explore freely his/her convictions and the set of nondominated solutions. The computational effort has been taken into account, as we have developed sensitivity analysis/re-optimization techniques to reduce the time needed to compute new nondominated solutions.

The purpose of this note is, therefore, to present the architecture of a DSS born from previous methodological work, complementing the papers [1,2] on the same research stream. In Section 2 the main characteristics of the DSS are presented, and an example is shown in Section 3. This note finishes with some conclusions in Section 4.

2. The main characteristics of the DSS

The DSS has been designed to solve MOILP/MOMILP problems. This system is based on interactive reference point procedures [1,2] mainly devoted to perform *directional searches* for nondominated solutions. At each interaction of a directional search, the DM needs only to select the objective function he/she wants to improve in relation to the previous nondominated solution. The procedure automatically adjusts the reference point and computes a new nondominated solution. Nothing else is necessary to abandon the previous solution and to compute a new one. The adjustment of the reference point is performed by sensitivity analysis, and leads to a parametric right-hand side scalarizing program which is solved using postoptimality techniques. So, *consecutive* nondominated solutions following particular trajectories (which improve one objective function at each interaction) can be obtained using this approach. These features promote *directional*

searches or *local searches*, since the following nondominated solution is *close* to the previous one.

The interactive procedure [1] was first developed for MOILP problems. This approach is based on cutting planes, and has shown considerable practical limitations. Nevertheless, it has also highlighted how much this type of search tool may be of value for a MOILP DSS. In fact, during a directional search the DM does not need to give explicitly new reference points which could lead to the same solution or to a solution distant from the current one.

Consequently, we have then developed the interactive procedure [2] which only differs from [1] in the technical approach to solve the scalarizing programs. This is based on the branch-and-bound methodology and applies not only to MOILP but also to MOMILP problems. It is intended to perform directional searches in an effective way by including an iterative routine of sensitivity analysis/re-optimization that uses information provided by the previous branch-and-bound tree to adjust automatically the reference point and to go ahead in the computation of the next nondominated solution.

The DSS also includes a procedure that combines the two previous approaches. It only applies to MOILP and combines cutting planes and branch-and-bound to solve the scalarizing programs and to perform the sensitivity analysis/re-optimization phases. The computational tests have shown that the branch-and-bound approach outperforms this one (because it is faster) in most problems, but there are certain types of problems (such as multiobjective multidimensional 0–1 knapsack problems) to which the combined approach suits better than the branch-and-bound approach.

In our opinion, *directional searches* provide a useful decision support tool for the DM. Nevertheless, the experience acquired from their application to several test problems raised some questions, such that: how useful are the directional searches in the initial phase of the decision process, when the DM does not know anything about the problem? If the DM wants to perform a global search, will it be easy for him/her to specify reference points explicitly?

We have felt that it is difficult to a DM to make a strategic search at an early phase of the decision process. A strategic search is important to get a holistic knowledge of the nondominated solution set of the problem, enabling the DM to define anchor points to the following search phases. A holistic knowledge of the problem can be provided by the computation of the nondominated solutions that optimize each objective function individually (and compose the pay-off table) and/or other supported nondominated solutions obtained by the optimization of weighted-sums of the objective functions considering a set of well-dispersed weighting vectors.

Another weak point of the *directional searches* is the following: in each interaction, the DM chooses an objective function to be improved, but he/she does not have control over the variation of the other objective functions. Then, the algorithm searches automatically for the *closest* solution in a predefined trajectory that improves the objective function selected by the DM. But the DM may also desire to control the variation of the other objective functions, namely by disallowing that they decrease below certain levels. Therefore, the procedure must give the DM the possibility of imposing additional limitations on the objective function values. It is worth noting that the introduction of additional limitations on the objective function values into the original *directional searches* corresponds to changing predefined trajectories to be followed. Thus, the DM is given a great control over the trajectory and does not require too much information about his/her preferences: just the objective function he/she wants to privilege at each moment, and lower bounds (optional) for some or all the other objective functions.

We have incorporated the above mentioned features into the DSS. Thus, the system offers the DM a set of tools, which can be used at any phase of the decision process. These tools aim at providing a progressive learning of the decision alternatives and a gradual establishment of the preferences of the DM. Some of them, such as the optimization of weighted-sums of the objective functions, are in general more useful in an initial phase of the decision process. The combination of the *directional searches* with the possibility of

imposing additional limitations on the objective function values can be used to scan nondominated solutions throughout different directions or to carry out a search focused on a delimited region (local search), for instance on the neighborhood of a nondominated solution that the DM considers interesting. The latter should be more useful in a final phase of the decision process.

The flowchart of Fig. 1 outlines the protocol of interaction with the DM implemented in the DSS.

The DSS has been implemented with the Delphi developer for Windows platforms. In the design of the user interface we have tried to make it easy to learn and easy to use. The basic components are a menu bar and a main window. The menus are intended to choose global operations, such as introducing/editing a problem or choosing an interactive multiobjective procedure to deal with the problem. The system also integrates metaheuristic approaches to deal with multiobjective 0–1 linear problems (the interested reader may refer to [3]). The problem files created by the DSS can be opened and changed using other programs, for instance the Microsoft Excel.

If the user selects an interactive multiobjective procedure to deal with the problem, the main window (Fig. 2 in the next section) will present tool bars with icon-controls that enable to perform specific operations, such as:

- to input information of preferences about the problem, namely to specify new reference points (information needed to perform *E* in Fig. 1), in order to choose an objective function to be improved during a directional search (to perform *F*), to specify weighting vectors for weighted-sums (to perform *C*) or to impose bounds on the objective functions (*D*);
- to demand the computation of new nondominated solutions, by projecting a reference point onto the nondominated solution set (*E*), starting a new *directional search* (entering *F*), continuing in the same directional search (keeping in *F*) or optimizing a weighted-sum (*C*); these computations can be performed considering the original feasible space or a restricted region delimited by bounds on the objective functions (*D*);

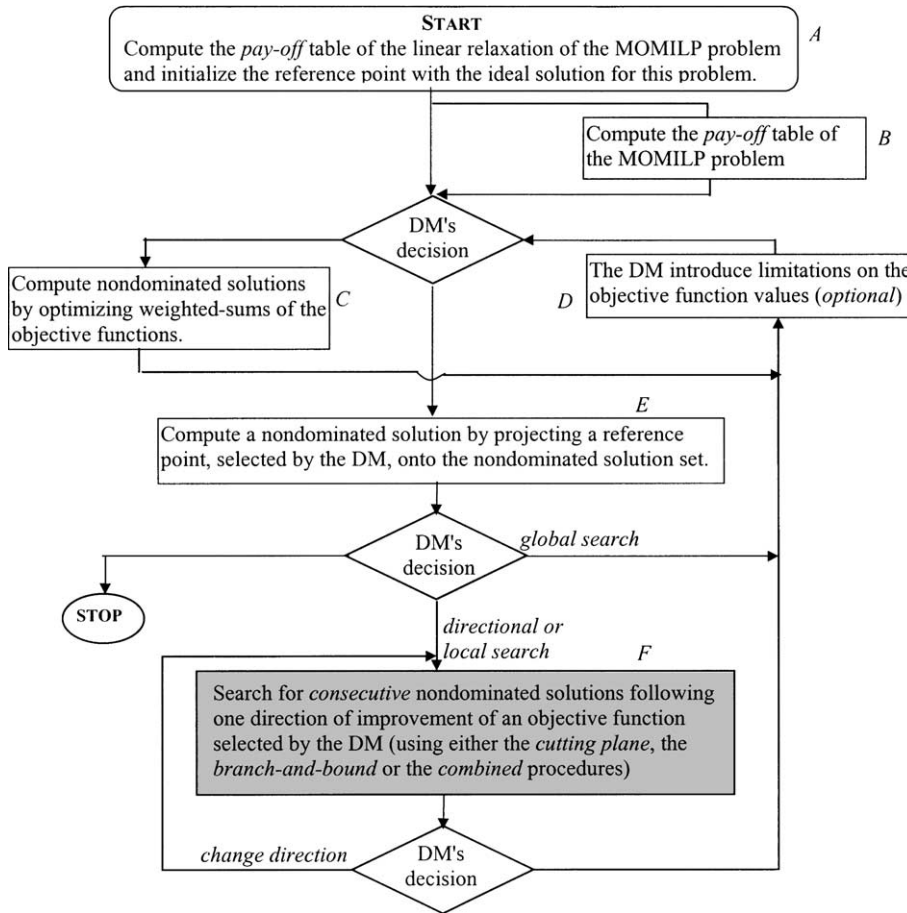


Fig. 1. Protocol to interact with the DM implemented in the DSS.

- operations that concern with the analysis of results: to configure some visualization characteristics (colors, or the number of solutions visible on the main window) or to open separate windows that show other numerical or graphical information; to save in disk, save/delete in/from the memory of the system or print nondominated solutions already computed.

The interface has been designed to suit an open communication approach with the DM(s). It enables a free exploration of the problem and a progressive learning of the set of nondominated solutions. There are no irrevocable decisions dur-

ing the whole process and the process only finishes when the DM(s) consider(s) to have gained sufficient insight into the nondominated solution set to base a final decision. More than one DM is allowed under the paradigm that they search for consensus, i.e. they work in a cooperative way. During the process, each DM can save/delete nondominated solutions in the memory of the system, keeping at the end of the procedure the solutions considered more satisfactory. The system enables a free exchange of information collected by the DMs. Each DM may choose a shade of colour for their preferred solutions, which is used to highlight the contrasts or convergence in the graphical representations.

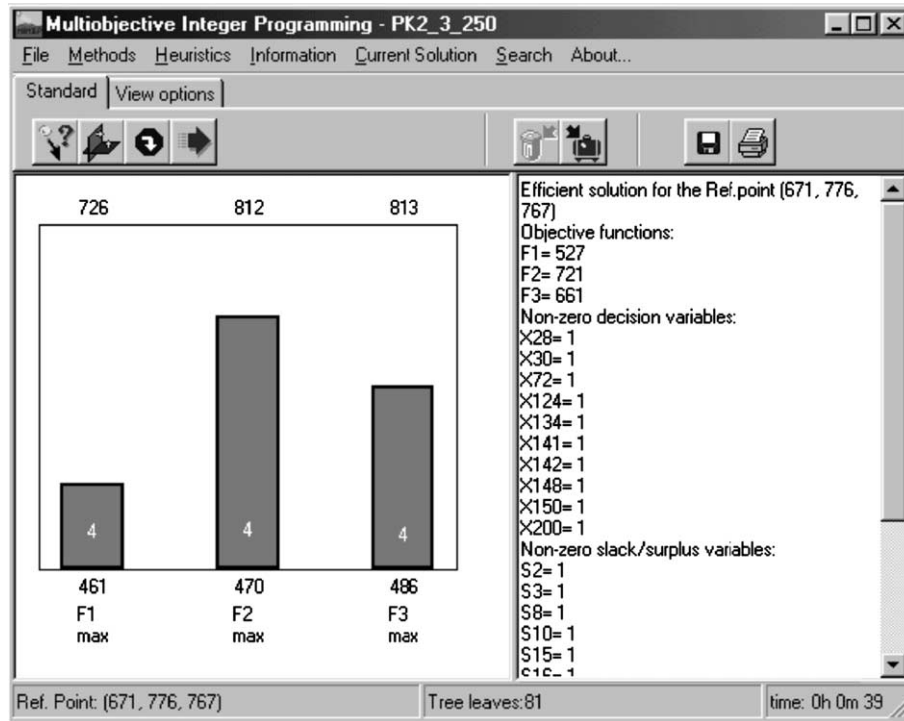


Fig. 2. The main window of the system showing solution 4.

3. An example

Let us consider a multiobjective integer problem with 250 binary variables, 50 constraints and three objective functions to be maximized. The structure of the constraints follows a packing problem, i.e. the constraint i is of the type $\sum_{j=1}^{250} a_{ij}x_j \leq 1$ with the a_{ij} 's equal to 0 or 1. In this example, the a_{ij} 's were randomly generated with the following probabilities (p):

$$\begin{cases} 0, & p = 0.75, \\ 1, & p = 0.25. \end{cases}$$

The coefficients of the objective functions were also randomly generated in the range of integers $[0, 30]$ (uniformly).

Suppose that the user wants to begin the decision process by knowing the nondominated solutions that optimize individually each objective function. So, he/she selects the corresponding option (computation of the solutions that make up the *pay-off* table). The solutions obtained are the

first three solutions saved in the memory of the system, whose objective function values are: $f^{(1)} = (671, 372, 401)$, $f^{(2)} = (473, 776, 633)$ and $f^{(3)} = (426, 545, 767)$. Thus, the ideal solution for this problem is $(671, 776, 767)$.

Then the user chooses the reference point approach based on branch-and-bound to explore the problem. In this case, the user is requested to choose a reference point to start the search. Consider that the ideal solution is chosen, so the reference point $(671, 776, 767)$ is projected onto the nondominated solution set. A new solution is computed, its objective functions vector being $f^{(4)} = (527, 721, 661)$. Fig. 2 shows the main window of the system with the solution 4. The main window includes three main panels: the panel on the left shows bar graphs for the last solution(s); the panel on the right shows the corresponding numerical information; above these, there is a panel of controls (divided into two tabs, "Standard" and "View options") which includes the main options available for this procedure.




Consider that the user wants to make a *directional search* (pressing button  of Fig. 2), and chooses criterion f_1 as the one to be improved. The next nondominated solution $f^{(5)} = (536, 625, 702)$ is obtained. Assuming that the user wants to *continue* the search in the same direction (pressing button ), then the nondominated solution $f^{(6)} = (542, 657, 589)$ is computed. If the users continues the search in the same direction, the next nondominated solution is $f^{(7)} = (604, 567, 657)$.

Fig. 3(a) shows the main window with bar graphs for the last four solutions ($f^{(4)}, f^{(5)}, f^{(6)}$ and $f^{(7)}$) and the numerical information of the last solution.

Fig. 3(b) shows *indifference regions* (for all the solutions already known) with respect to the reference points. An *indifference region* means a region (set) of reference points that lead to the same solution. The space of the reference points coincide with the space of the objective functions. So, when this space is of dimension 3 (or 2), it can be graphically visualized, and no information is lost if we consider just a cut, i.e. a plane (line) where the sum of the components of the reference points is constant. Further, the visualization can be re-

duced to a triangle (line segment) inside this plane (line) provided that the limits are defined properly. All the nondominated solutions of the problem have an indifference region on this triangle (line segment), and the individual optima for the objective functions fill areas near the vertices of the triangle (line segment). This can be observed in Fig. 3(b) where solutions 1, 2 and 3, which optimize f_1, f_2 and f_3 , respectively, appear in the corners of the triangle. The indifference regions on the reference point space are generally nonconvex and difficult to determine. So, at each interaction, the DSS only determines convex sub-regions for one or more nondominated solutions (instead of the whole indifference region). Then, the DSS fills the corresponding areas in the triangle with the colours assigned to those solutions in other graphs. These areas give information to the users about reference points that are uninteresting to be selected because they surely lead to solutions already known. For more details on this topic, see [4].

Suppose that now the user decides to impose the limitation $f_2 \geq 580$ (by choosing option ) and selects a new starting reference point (pressing

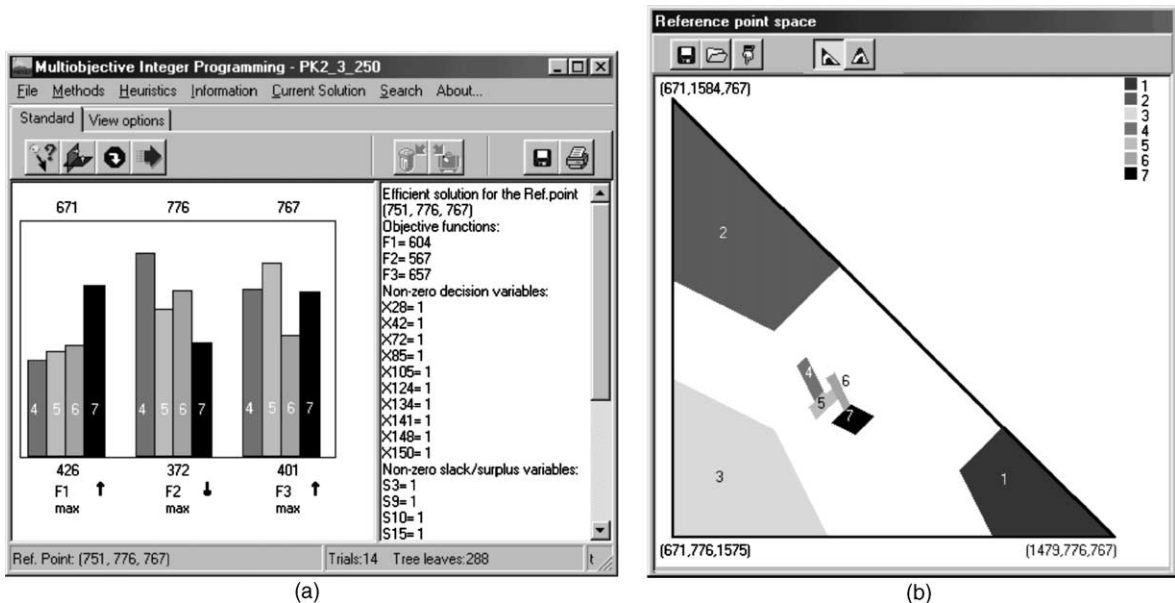


Fig. 3. Graphs after computing solution 7. (a) Representation on the main window. (b) Indifference sub-regions on the reference point space.

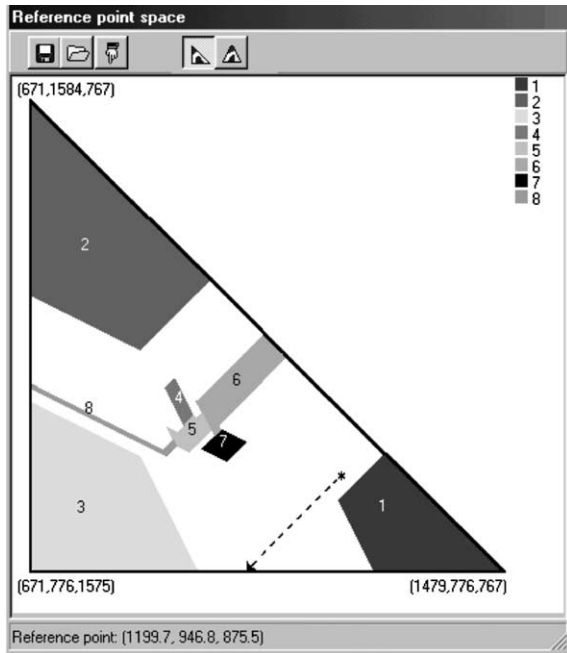


Fig. 4. Indifference sub-regions for the solutions computed.

of Fig. 3(a) and inputting the values of the new reference point into a dialog box, or just clicking on the desired point of the triangle of Fig. 3(b)). Let the point marked with an * in Fig. 4 be the new reference point selected by the user. This reference point is projected onto the nondominated solution set restricted by $f_2 \geq 580$, leading to the already known solution 6: $f^{(6)} = (542, 657, 589)$. Another indifference sub-region of solution 6 (on the reference points space) is computed and attached to the area previously determined for this solution. This can be observed if we compare Fig. 4 with Fig. 3(b). Note that the region of solution 6 in Fig. 4 does not include the selected reference point marked with an *. This situation is only possible because the reference point was projected onto the nondominated solution set restricted by the limitation $f_2 \geq 580$. Thus, this reference point would certainly lead to a solution that does not satisfy $f_2 \geq 580$ if it was projected onto the whole nondominated solution set.

Then, a new *directional search* is initialized in the region restricted by $f_2 \geq 580$. Starting from the current point, the user selects f_3 to be im-

proved. The next solution obtained is $f^{(5)} = (536, 625, 702)$, which is already known. *Continuing* the search in the same direction, the next nondominated solution is $f^{(8)} = (504, 596, 756)$. If the user wants to *continue* the search in the same direction, he/she is then informed that the optimum of f_3 was reached in the region restricted by $f_2 \geq 580$. So, the decision process could proceed in another direction or from a different starting reference point.

Fig. 4 illustrates the previous search. The dashed line in this figure shows the trajectory of reference points that would be followed if the limitation $f_2 \geq 580$ had not been included. As we have noticed before, the introduction of additional limitations on the objective function values during *directional searches* corresponds to changing the predefined original trajectories.

4. Conclusions

A new DSS devoted to multiobjective integer and mixed-integer linear programming models was introduced in this note. It proposes a learning oriented search for the nondominated solution set rather than the convergence to any implicit utility function of the DM. We believe in open communication decision support tools combining two search phases. First, a strategic phase dedicated to getting a rough idea of the shape of the nondominated solution set, in order to provide information that enables the reduction of the scope of the search, namely the narrowing the feasible region by introducing bounds on the objective function values. In a final phase a free local search is recommended. Although, in most of the situations, this is the normal sequence of learning the nondominated solution set, the procedures of the different phases can be mixed, i.e. they can be combined interactively.

The use of graphics and interactive tools to introduce and/or to manage data and to visualize results were extensively explored in this DSS, taking into account the cognitive strengths and the limitations of the Human Being. Nevertheless, the major innovation of this work is related to the originality of the calculation tools, namely in the directional/local search phases. The elicitation of

information from the DM to pursue the search is not too hard, i.e. he/she is just asked about the progressive introduction of bounds on the objective function values, taking into account the current state of knowledge of the nondominated solution set, and of the objective function he/she wants to improve to follow a directional/local search. Furthermore, it concerns exclusively objective function values, as recommended by several psychological studies on interactive procedures. Some features of the DSS were illustrated using an example.

5. Final notes

The distribution of this software is free. The software and/or the complete formulation of the example in Section 3 can be obtained from the authors.

It is worth noting that this software can be useful for educational purposes. First, because it enables a progressive learning of the problem and a free exploration of the set of nondominated solutions. Second, because the graphics provided by this system, namely the representation of indifference regions of the reference point space (available for problems with two or three objective functions), make the software useful for a more technical teaching. The interpretation of this graph gives a deeper knowledge of the operating mode of

reference point approaches when dealing with discrete sets of nondominated solutions.

References

- [1] M.J. Alves, J. Clímaco, Using cutting planes in an interactive reference point approach for multiobjective integer linear programming problems, *European Journal of Operational Research* 117 (3) (1999) 565–577.
- [2] M.J. Alves, J. Clímaco, An interactive reference point approach for multiobjective mixed-integer programming using branch-and-bound, *European Journal of Operational Research* 124 (3) (2000) 478–494.
- [3] M.J. Alves, J. Clímaco, An interactive method for 0–1 multiobjective problems using simulated annealing and tabu search, *Journal of Heuristics* 6 (3) (2000) 385–403.
- [4] M.J. Alves, J. Clímaco, Indifference sets of reference points in multiobjective integer linear programming, *Journal of Multi-Criteria Decision Analysis* 10 (4) (2001) 177–189.
- [5] C. Ferreira, B.S. Santos, M.E. Captivo, J. Clímaco, C.C. Silva, Multiobjective location of unwelcome or central facilities involving environmental aspects: A prototype of a decision support system, *Belgian Journal of Operations Research, Statistics and Computer Science* 36 (2–3) (1996) 159–172.
- [6] W. Ogryczak, K. Studzinski, K. Zorychta, DINAS: A computer-assisted analysis system for multiobjective transportation problems with facility location, *Computers and Operations Research* 19 (7) (1992) 637–647.
- [7] V. Vassilev, S. Narula, P. Vladimirov, V. Djambov, MOIP: A DSS for multiple objective integer programming problems, in: J. Clímaco (Ed.), *Multicriteria Analysis*, Springer, Berlin, 1997, pp. 259–268.